Application No: 10/706,516
Office Action mailed: August 27, 2008
Reply dated: November 26, 2008

Remarks

The above Amendments and these Remarks are in reply to the Office Action mailed August 27, 2008 and an interview with the Examiner on November 26, 2008.

Applicant acknowledges with thanks Examiner Nguyen's assistance in granting an interview with Thomas Plunkett on November 26, 2008, during the course of which interview the participants generally discussed the claimed embodiments. Applicant agreed to further amend Claim 1 to more clearly define the embodiment therein. Examiner Nguyen indicated that further consideration of the application would follow upon receipt of this Reply.

I. Summary of Rejections

In the Office Action, claims 1, 12, 16, 17, 26, and 27 were rejected under 35 U.S.C. 102(b) as allegedly anticipated by Bentley et al. (U.S. Patent No. 5,815,415, hereinafter Bentley). Claims 10 and 18 were rejected under 35 U.S.C. 103(a) as allegedly unpatentable over Bentley in view of Click, Jr. et al (U.S. Patent No. 6,381,737, hereinafter Click, Jr.). Claim 11 was rejected under 35 U.S.C. 103(a) as allegedly unpatentable over Bentley in view of Glynias et al. (U.S. Patent No. 6,125,383, hereinafter Glynias). Claim 13 was rejected under 35 U.S.C. 103(a) as allegedly unpatentable over Bentley in view of Official Notice. Claim 25 was rejected under 35 U.S.C. 103(a) as allegedly unpatentable over Bentley in view of Whitehead et al. (U.S. Patent No. 6,085,030, hereinafter Whitehead).

II. Summary of Applicant's Response

The present Reply amends claim 1 and 10; and adds new claims 28-36; leaving for the Examiner's present consideration claims 1, 10-13, 16-18, and 25-36. Reconsideration of the claims is requested.

III. 35 U.S.C. 102(b) Rejections

Claim 1

Claim 1 (as amended) defines:

1. A computer program product including a storage medium with instructions thereon for execution by a computer for high level dynamic code generation, the instructions comprising:
   a) computer code for dynamically generating at run-time a class file container object that stores source code describing a class and selecting a class name and a super class for the class;
   b) computer code for adding a first source code defining a method to the class stored in the class file container object, wherein an application programming interface is used to define the method;
   c) computer code for repeating step b for each method of the class;
   d) computer code for adding a second source code into the method in the class stored

- 6 -

Attorney Docket No.: ORACL-01316US2
tplunkett/oracl/1316us2/1316us2 Reply 082708.doc

PAGE 7/10 * RCVD AT 11/26/2008 8:20:14 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-6/11 * DNIS:2738300 * CSID:7033590094 * DURATION (mm-ss):02-08

in the class file container object, wherein an application programming interface is used to define code added into the method;

    e) computer code for repeating step d to populate each method of the class stored in the class file container object;

    f) computer code for generating a tree of statements and expressions based on the class stored in the class file container object, wherein each statement and expression is represented as an object, wherein each statement maintains state of the program being generated;

    g) computer code for using the tree of statements and expressions to generate byte code for the class; and

    h) computer code for instantiating an instance of the new class file object from the byte code.

Claim 1 was rejected under 102(b) as allegedly anticipated by Bentley. Applicant notes that in the prior response there was a discrepancy between the version of Claim 1 with the Listing of the Claims and the version of Claim 1 with the remarks. Applicant regrets any confusion caused by the discrepancy.

Claim 1 has been amended to define dynamically generating at run-time a class file container object that stores source code describing a class that stores source code describing a class and selecting a class name and a super class for the class. In the Office Action, it was asserted that Bentley discloses dynamically generating at run-time a class file container object that stores source code describing a class.

Bentley discloses that a platform-dependent Computerized Modeling System (CMS) includes a static kernel and a dynamic framework. The static kernel executes on the platform and interfaces to the operating system and the computer hardware, and provides services necessary to load and execute CMS services and to interface to the platform services. The dynamic framework executes on the platform and interfaces to the kernel, providing a platform-independent visual interface between the CMS and a CMS user, and employs the services of the kernel. (Col. 6, lines 3-12). Bentley further discloses that in terms of the run-time CMS, a class is part of a schema and defines an object that may be instantiated from the class, where the instantiated object can be placed in the model. (Col. 12, lines 36-40). Bentley further discloses that the interface 'Opening' has the methods 'setDimensions', 'getHeight', and 'getWidth.' Any class declared as supporting the interface opening must implement all three of these methods with the signatures defined in the interface declaration. At run-time, the interface declaration is embodied in an interface object for the interface. (Col. 14, lines 6-13).

Bentley further discloses that the CMS supports the automatic reading and writing of all data types by the object/persistence manager of the CMS kernel. Nevertheless, it may be the case that certain "C" data types such as unions, bitfields, and non-object pointers are not supported. Accordingly, if a class introduces unsupported member variable types, such a class must also provide methods for serializing its state. (Col. 42, lines 27-34). Bentley further discloses that in the programming language 53, a class declaration defines a class 54 by listing the member variables 60 and methods 62 for the class 54. (Figure 5, Col. 12, lines 51-53).

- 7 -

Attorney Docket No.: ORACL-01316US2
tplunkett/oracl/1316us2/1316us2_Reply_082708.doc
PAGE 8/10 * RCVD AT 11/26/2008 8:20:14 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-6/11 * DNIS:2738300 * CSID:7033590094 * DURATION (mm-ss):02-08

Bentley further discloses an example of the class declaration 64 of StormDoor. (Figures 6-7, Col. 13, lines 28-38). Bentley further discloses that a template 58 provides a flexible programming technique for declaring classes 54 and interfaces 56. A template 58 is a declaration of a "shell class" or "shell interface" with some of the type information parameterized (i.e. not filled in). The parameterized type information is then filled in during compile time according to other source code. (Col. 14, line 64 – Col. 15, line 2).

Applicant respectfully submits that the framework disclosed by Bentley is dynamic because it is platform-independent, in contrast to the static kernel which is platform-dependent. Applicant respectfully submits that the object disclosed in Bentley (Col. 12, lines 36-40) that is instantiated at run-time and the interface object disclosed in Bentley (Col. 14, lines 6-13) do not store source code that defines a class. Instead, Applicant respectfully submits that Bentley discloses run-time objects that do not contain source code. Applicant respectfully submits that Bentley discloses that a class is part of a schema and defines an object that may be instantiated from the class, however Bentley does not appear to disclose a class file container object that stores source code describing a class. Applicant respectfully submits that the computerized modeling system disclosed in Bentley (Col. 42, lines 27-34) uses an object/persistence manager to read and write state information for data types using serialization, or alternative methods for unsupported data types. Applicant respectfully submits that the parameterized type information is filled in at compile time, and is not dynamically generated at run-time.

Therefore, Applicant respectfully submits that Bentley is silent with regards to dynamically generating at run-time a class file container object that stores source code describing a class.

Claim 1 further defines generating a tree of statements and expressions based on the class stored in the class file container object, wherein each statement and expression is represented as an object, wherein each statement maintains state of the program being generated.

Bentley discloses block diagrams showing parse trees developed by a compiler. (Figures 25 and 26).

Applicant respectfully submits that Bentley is silent regarding generating a tree of statements and expressions based on the class stored in the class file container object, wherein each statement and expression is represented as an object, wherein each statement maintains state of the program being generated.

In view of the comments provided above, Applicant respectfully submits that the embodiment defined by Claim 1, as amended, is neither anticipated by, nor obvious in view of the cited references, and reconsideration thereof is respectfully requested.

Claims 12, 16, 17, 26, and 27

Claims 12, 16, 17, 26, and 27 depend from and include all of the features of Claim 1. Claims 12, 16, 17, 26, and 27 are not addressed separately, but it is respectfully submitted that these claims are allowable as

- 8 -

Application No: 10/706,516
Office Action mailed: August 27, 2008
Reply dated: November 26, 2008

depending from an allowable independent claim, and further in view of the amendments to the independent claim, and the comments provided above. Reconsideration thereof is respectfully requested.

IV.   35 U.S.C. 103(a) Rejections
Claims 10-11, 13, 18, and 25

Claims 10-11, 13, 18, and 25 depend from and include all of the features of Claim 1. Claims 10-11, 13, 18, and 25 are not addressed separately, but it is respectfully submitted that these claims are allowable as depending from an allowable independent claim, and further in view of the amendments to the independent claim, and the comments provided above. Reconsideration thereof is respectfully requested.

V.   Additional Amendments

Claims 28-36 have been newly added by the present Reply. Applicant respectfully requests that new claims 28-36 be included in the application and considered therewith.

VI.   Conclusion

In light of the above, it is respectfully submitted that all of the claims now pending in the subject patent application should be allowable, and reconsideration thereof is respectfully requested. The Examiner is respectfully requested to telephone the undersigned if he can assist in any way in expediting issuance of a patent.

The Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 06-1325 for any matter in connection with this response, including any fee for extension of time, which may be required.

Respectfully submitted,

Date: November 26, 2008   By: /Thomas K. Plunkett/
                                             Thomas K. Plunkett
                                             Reg. No. 57,253

Customer No. 80548
FLIESLER MEYER LLP
650 California Street, 14th Floor
San Francisco, California 94108
Telephone: (415) 362-3800
Facsimile: (415) 362-2928

- 9 -

Attorney Docket No.: ORACL-01316US2
tplunkett/oracl/1316us2/1316us2 Reply 082708.doc

PAGE 10/10 * RCVD AT 11/26/2008 8:20:14 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-6/11 * DNIS:2738300 * CSID:7033590094 * DURATION (mm-ss):02-08